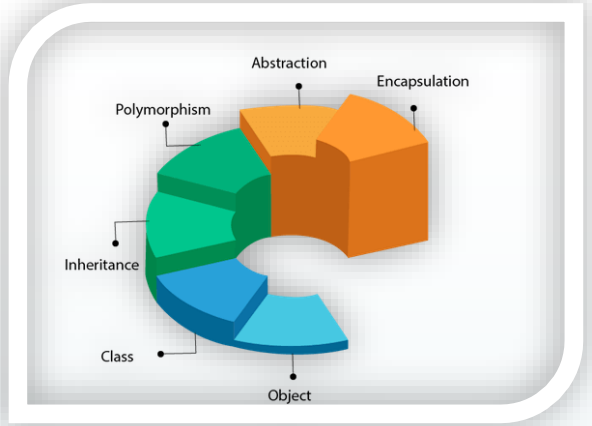
YouTube BM DERSLERİ

http://youtube.com/bmdersleri

MAKÜ

BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ

Nesneye Dayalı Programlama Dersi



JAVA'DA KARAR YAPILARI

Hazırlayan ve Sunan: **HİLAL ÇINAR - 1911404044**

Tarih : 11/06/2021

Sürüm : v2

Ders Yürütücüsü : Doç. Dr. İsmail KIRBAŞ

MAKÜ
BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ

İçindekiler

Karar Verme Nedir?
Karar Verme Aşamaları Nelerdir?

1-Karar ihtiyacının ortaya çıkması

2-Problemin tanımlanması

3-Alternatiflerin belirlenmesi ve irdelenmesi

4-Alternatifler arasından seçim yapılması

5-Kararın uygulanması

6-Geri bildirim alınması

Java'da Karar Yapıları Nelerdir?

If

If / else

Nested if

Switch – Case

If / else – if

Java'da If ifadesi

En basit karar verme ifadesidir.

If karar yapısının cümle içerisinde kullanımına örnek verecek olursak;

"Eğer Hava yağmurlu ise şemsiyeni al."

"Eğer Aldığın Not 50'den fazla ise sınavı geçtin."

If ifadesinin genel biçimi veya sözdizimi şöyledir:

```
if (koşul) {
   // Yürütülmek istenen ifadeler (Çalıştırılmak istenen kod)
}
```

Koşul durumu boolean veri tipinde olmalıdır.

Java'da If ifadesi

If ifadesinin genel biçimi veya sözdizimi şöyledir:

```
if (koşul) {
   // Yürütülmek istenen ifadeler (Çalıştırılmak istenen kod)
}
```

Koşul durumu boolean veri tipinde olmalıdır.

If ifadesinden sonra süslü küme parantezleri {} sağlamazsak, o zaman varsayılan olarak if ifadesinden sonraki tek ifadeyi dikkate alır.

Örneğin;

```
if (koşul)
     ifade1;
     ifade2;
```

Bu durumu kodlar üzerinden inceleyecek olursak;

Java'da If ifadesi

If ifadesinin genel biçimi veya sözdizimi şöyledir:

```
if (koşul) {
   // Yürütülmek istenen ifadeler (Çalıştırılmak istenen kod)
}
```

Koşul durumu boolean veri tipinde olmalıdır.

If ifadesinde koşuldan sonra noktalı virgül konulmamalıdır

Bu durumu kodlar üzerinden inceleyecek olursak;

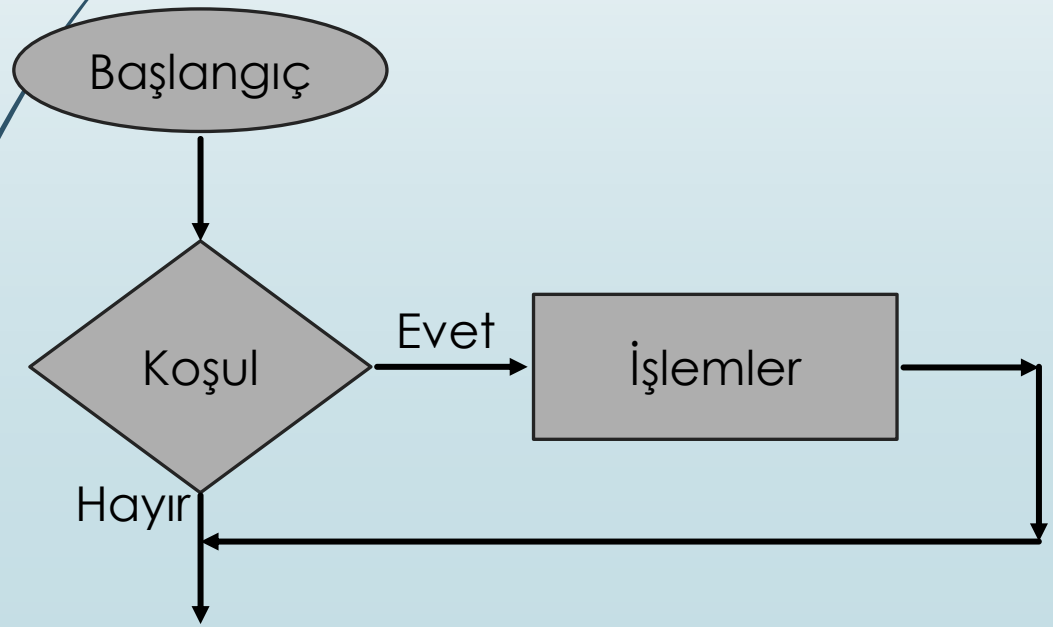
Java'da If ifadesi

If ifadesinin genel biçimi veya sözdizimi şöyledir:

```
if (koşul) {
   // Yürütülmek istenen ifadeler
(Çalıştırılmak istenen kod)
}
```

Koşul durumu boolean veri tipinde olmalıdır.

If ifadesinin genel akış şeması şöyledir:



If yapısına tek seçimli yapı denilmektedir.

If yapısı için örnek bir kod yazacak olursak;

Java'da If / else ifadesi

Koşulun yanlış olduğu durumda başka bir kod bloğu yürütmek istersek?

If / else ifadesinin genel biçimi veya sözdizimi şöyledir:

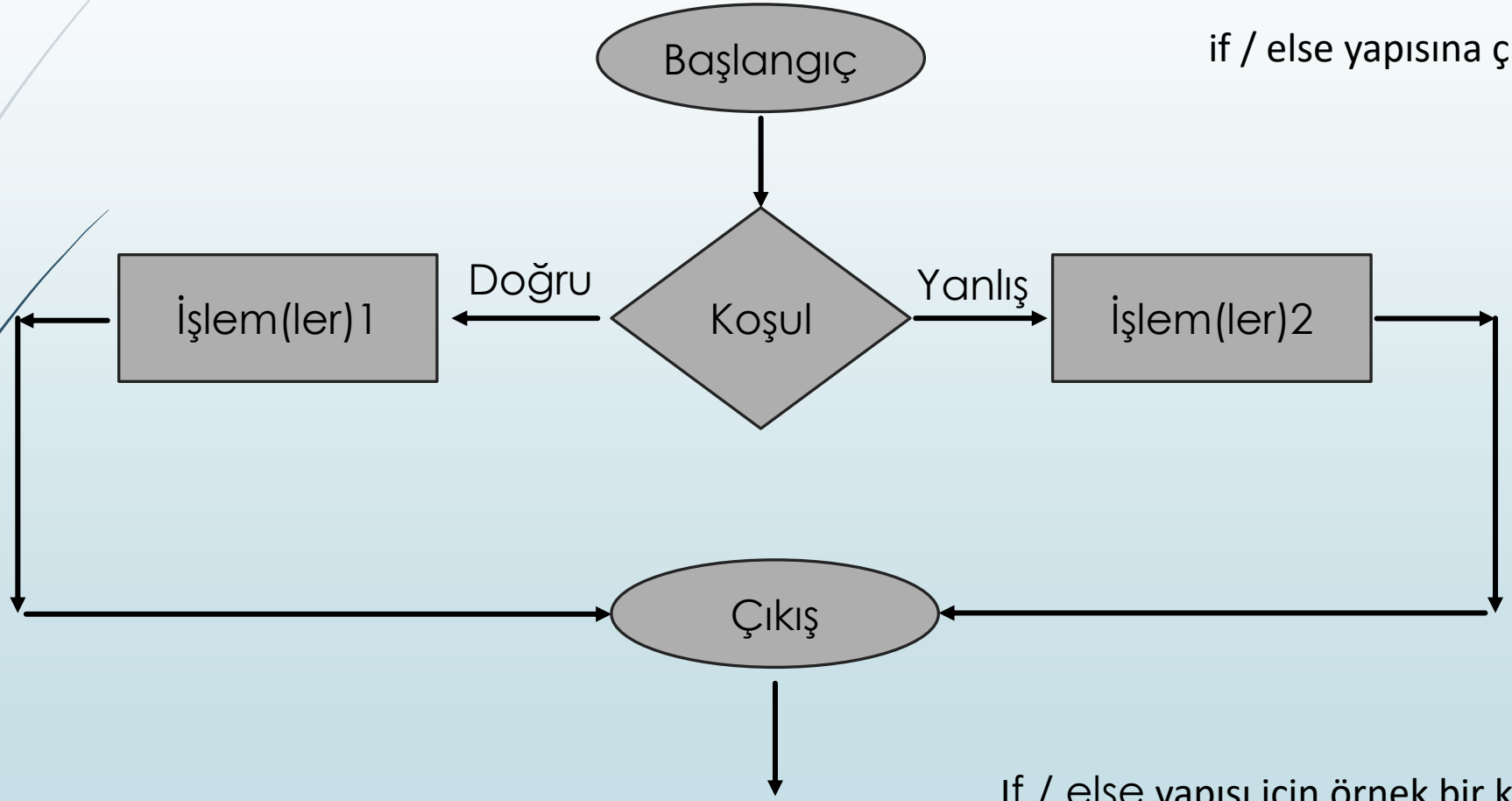
```
if (koşul)
{
    // Koşul doğru ise bu süslü
parantezler arasında verilen
kodları yürütür
}else{
    // Koşul yanlış ise bu süslü
parantezler arasında verilen
kodları yürütür
}
```

If ifadesinde olduğu gibi if / else ifadesinde de süslü parantezlerin kullanımı önemlidir.

Bu durumu kodlar üzerinden inceleyecek olursak;

Java'da If / else ifadesi

If / else ifadesinin genel akış şeması şöyledir:



if / else yapısına çift seçimli yapı denir.

If / else yapısı için örnek bir kod ve akış diyagramı örneği yazacak olursak;

Java'da Ternary Operatörü

Ternary Operatörü kendi kalıbında şarta göre farklı değerleri döndürmemizi sağlayan bir operatördür.

```
boolean cinsiyet

                    → Kadınlara özel ürünler için kampanya
String mesaj
                    → Erkeklere özel ürünler için kampanya
```

```
String mesaj =..Şart/Durum..?..Şart doğru olduğunda..:..Şart Yanlış olduğunda..
```

Java'da iç içe geçmiş if ifadesi (Nested if)

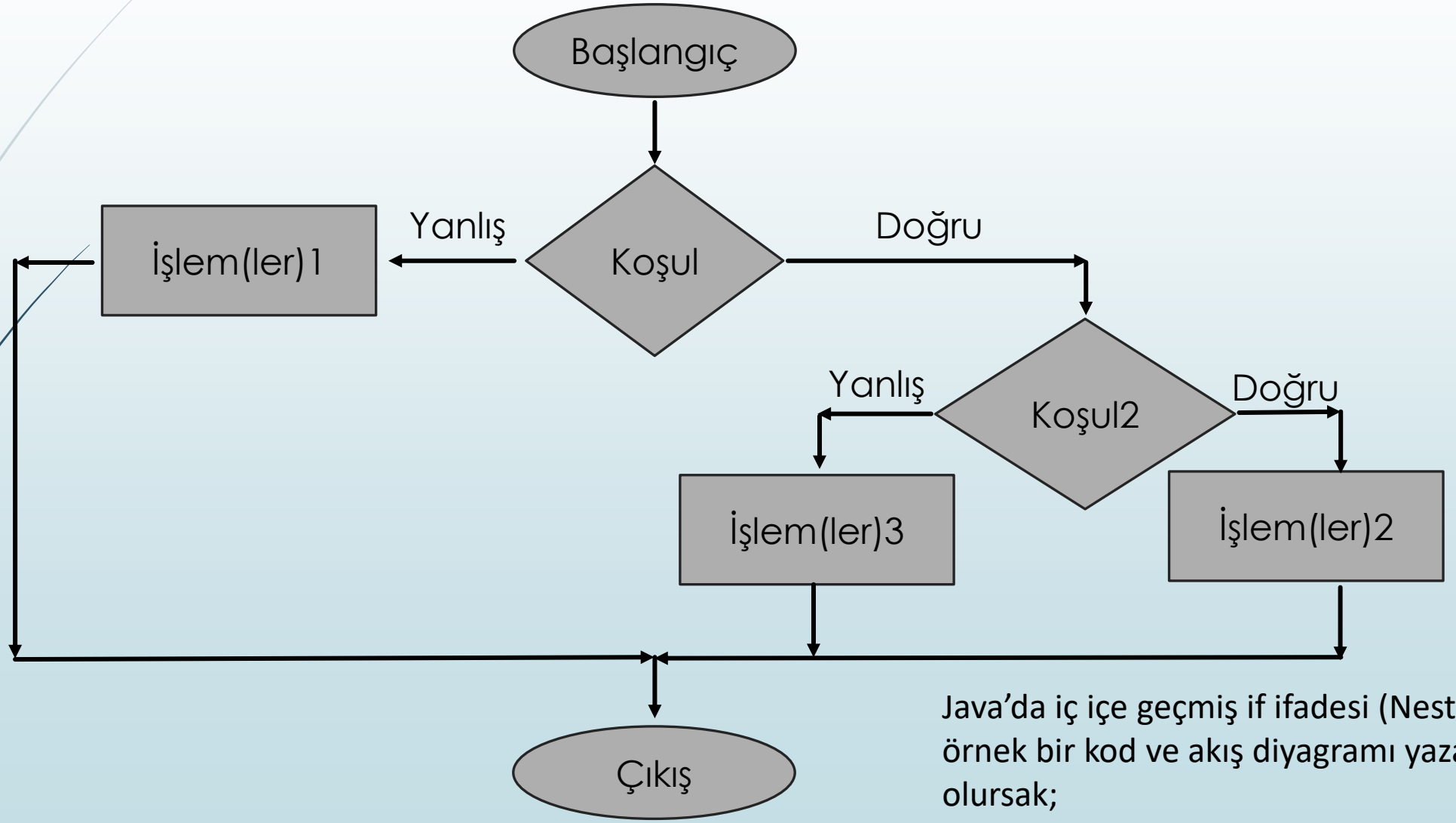
İç içe geçmiş if, if'in gövdesinde veya başkasının gövdesinde başka bir if ifadesidir. Java, yuvalanmış if ifadelerine izin verdiği için, bir if veya else-if ifadesini başka bir if ifadesinin içine yerleştirebiliriz.

Java'da iç içe geçmiş if ifadesinin (Nested if) genel biçimi veya sözdizimi şöyledir:

```
if (koşul1)
{
   // koşul1 doğru olduğunda
çalıştırılır
   if (koşul2)
   {
      // koşul2 doğru
olduğunda çalıştırılır
   }
}
```

Java'da iç içe geçmiş if ifadesi (Nested if)

Java'da iç içe geçmiş if ifadesinin (Nested if) *genel akış şeması şöyledir:*



Java'da iç içe geçmiş if ifadesi (Nested if) için örnek bir kod ve akış diyagramı yazacak olursak;

Java’da if / else – if ifadesi

If-else-if merdiveni, Java'da çok yaygın bir programlama yapısıdır ve görünüşü nedeniyle if-else-if merdiveni olarak da adlandırılır.

Bu şekilde bir yazımda, kullanıcı birden fazla seçenek arasından karar verebilir.

Ayrıca if, elseif veya else'den sonra çalışması gereken kod bir tane ise { } parantezleri kullanmak da gerekmez.

Bu durumu kodlar üzerinden inceleyecek olursak;

Java'da if / else – if ifadesi

if / else – if ifadesinin *genel biçimi veya sözdizimi şöyledir:*

```
if(koşul1)
{
//koşul1 doğru olduğunda çalıştırılır
}
else if (koşul2)
{
//koşul2 doğru olduğunda çalıştırılır
}
else if (koşul3)
{
//koşul3 doğru olduğunda çalıştırılır
}
.
.
.
else
{
//Koşullardan hiçbiri doğru değilse çalışacak kod kısmı
}
```

Java'da if / else – if ifadesi için örnek bir yazacak olursak;

Java'da Switch – Case ifadesi

Switch - case yapısı değişkenin farklı durumları için farklı işlemler yapar.

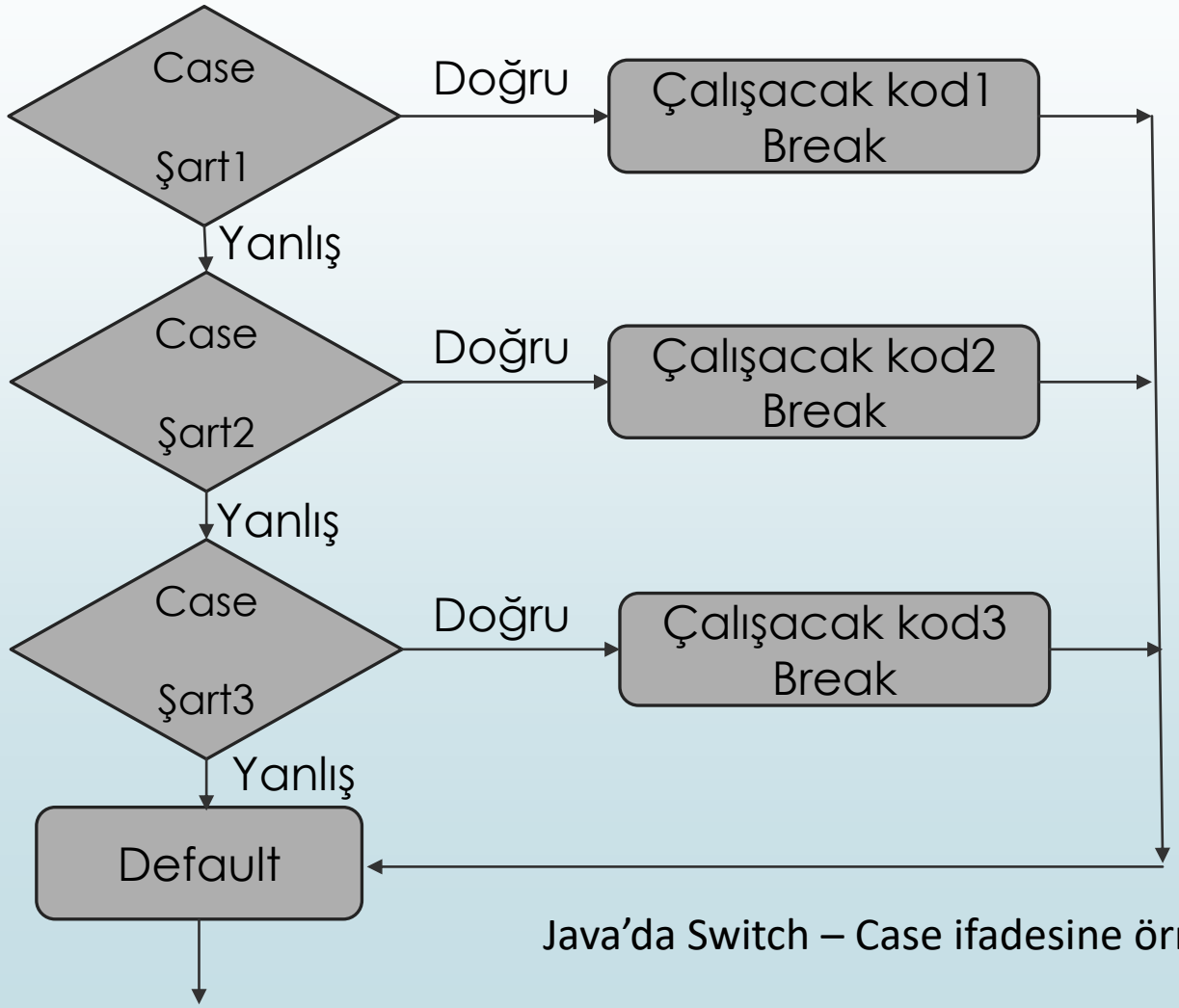
Java'da Switch – Case ifadesinin genel biçimi veya sözdizimi şöyledir:

```
switch (Değişken ismi) {
        case koşul1:
           Çalıştırılacak kod
           break;
        case koşul2:
           Çalıştırılacak kod
           break;
        case koşul3:
           Çalıştırılacak kod
           break;
        default:
           Çalıştırılacak kod
    }
```

```
switch (Değişken ismi) {
        case koşul1:
           Çalıştırılacak kod
           break;
        case koşul2:
           Çalıştırılacak kod
           break;
        case koşul3:
         case koşul4:
           Çalıştırılacak kod
           break;
        default:
           Çalıştırılacak kod
    }
```

Java'da Switch – Case ifadesi

Java'da Switch – Case ifadesinin genel akış şeması şöyledir:



Java'da Switch – Case ifadesine örnek kod yazacak olursak;

Break – Continue nedir?

Break, döngüyü sona erdirir, bu ne anlama gelir programın akışında herhangi bir yerde break gördüğümüz de break görür görmez içinde bulunduğumuz döngü anında koşula bakmadan sona erer koşulun doğru ya da yanlış olması hiçbir şey ifade etmez.

Continue, altındaki işlemleri yapmadan döngünün en başına gider... Örneğin programın herhangi bir yerinde continue görüldü continue direkt olarak döngünün en başına gider ve alttaki hiçbir işlemi gerçekleştirmez.

Continue kısmına kod ekranında bakacak olursak;

Sonuç

Şuana kadar incelediğimiz karar durumlarını sırasıyla özetleyecek olursak;

If durumu için; if durumunda girilen koşulumuz doğru olarak değerlendirilirse, koşul doğru olduğunda çalışması istenilen kod çalışır, girilen koşul false dönüyorsa if durumunda herhangi bir geri dönüş olmaz. Ve sonlanır. Çünkü tek seçimli bir yapıdır.

If / else durumunda; artık if durumundan farklı olarak çift seçimli bir yapı vardır. Yani koşulumuzun yanlış olduğu durum da çalıştırabileceğimiz bir kod bloğu daha vardır. Koşul true olarak döndüğünde if kısmının altında bulunan durum çalışır koşul false şeklinde döndürüldüğündeyse else durumunun altında bulunan kod bloğu çalışır.

Sonuç

Ternary Operatörü için; Ternary Operatörü kendi kalıbında şarta göre farklı değerleri döndürmemizi sağlayan bir operatördür. Bu Operatör tam olarak if - else operatörleri ile yapılan işlemleri tek satırda yapmamızı sağlar. Mantığı; ilk önce bir şart durumuna sahip olmamız gerektiği ile başlar, daha sonra Ternary operatörünün kalıbına uyarak soru işareti ve iki nokta üst üste kullanırız. İki nokta üst üstenin iki yanında boşluklar bulunur ve bu boşluklardan birine koşul doğru olduğunda çalışacak kodu diğerineyse koşul yanlış olduğunda çalışacak kodu yazarız. Bu operatörü her durumda kullanmak mantıklı değildir ama bazı durumlarda çok işimize yaramakta temiz kod yazmamızı sağlamaktadır.

Nested if kısmına baktığımızda; iç içe geçmiş bu if'ler en dıştaki if'in koşulunun true dönmesi ile çalışma ihtimali olan if durumlarıdır. Bu if'lerin de çalışabilmesi için o if'ler için girilen koşulların doğru olması gerekir. İç kısımda bulunan if'lerin koşullarının doğru ya da yanlış olması dışta bulunan if yapısının çalışıp çalışmama durumunu etkilemez.

Sonuç

If / else – if durumu için; bu durumda elimizde pek çok ihtimal bulunmaktadır. Pek çok durumun olabilmesine ihtimal sağlar. İlk kısımda bulunan if durumunun koşulu false dönerse yukarıdan aşşağı else if durumlarının koşulları kontrol edilir ve true dönen bir koşul olana kadar değerlendirme devam eder eğer true dönmüyorsa en sonda bulunan else durumu çalışacaktır.

Switch – case durumuna gelecek olursak; if / else – if durumuna benzer, ama Switch – case durumunda aralıklarla değil nokta atışlarıyla çalışırız. İstediğimiz durumu direkt olarak belirtmemiz gereklidir. Bu durum da yukarıdan aşşağı bir mantık ile çalışır ve her Switch – case ifadesinde break komutunun bulunması gereklidir yoksa Switch – case 'e özgü olan bir durum ile bir kere true döndüğünde default kısmı da dahil tüm case'lerin kod bloklarını çalıştıracaktır. Hiç bir case true dönmez ise en sonunda default ile bitirilir.

Break, döngüyü sona erdirir ve Continue, altındaki işlemleri yapmadan döngünün en başına gider…

MAKÜ
BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ

Kaynaklar

- **Genel Karar Yapıları Anlatımı İçin;**
- https://hakankanmaz.wordpress.com/2015/02/22/javada-kontrol-yapilari-control-structures/
- https://peakup.org/blog/if-karar-yapisi-ile-sarta-gore-islemler/
- https://www.geeksforgeeks.org/decision-making-javaif-else-switch-break-continue-jump/
- https://techvidvan.com/tutorials/decision-making-in-java/
- https://data-flair.training/blogs/decision-making-in-java/
- https://emrekosedotcom.wordpress.com/2014/09/28/javada-operatorler-ve-karar-yapilari/
- http://bilgisayarbilim.com/karar-mantik-yapisi-ile-problem-cozme/
- https://phpdefteri.com/icerik/16/ifelse_kontrol_yapisi.html
- https://emrecelen.com.tr/javada-kontrol-yapilari/
- https://ramazanbiyikci.com.tr/java-karar-yapilari-if-else-if/

YouTube BM DERSLERİ

http://youtube.com/bmdersleri

MAKÜ
BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ

Kaynaklar

- **Java'da Switch - Case Yapısı Anlatımı İçin;**
- https://www.kodkampusu.com/javada-switch-case-yapisi/
- https://kayademirli.com/javada-switch-case/
- https://ramazanbiyikci.com.tr/java-karar-yapilari-switch-case/
- https://www.sahinsezgin.com/csharp-karar-yapilari-switch-case/
- https://muratyagis.net/javascript-switch-case/

- **Farklı Bakış Açısı İçin C# Karar Yapıları Kullanımı:**
- https://www.algoritmaornekleri.com/c-sharp/c-if-else-kullanimi/
- https://www.algoritmaornekleri.com/c-sharp/c-karar-yapilari/
- https://www.algoritmaornekleri.com/c-sharp/c-ic-ice-if-kullanimi/

YouTube BM DERSLERİ

http://youtube.com/bmdersleri

MAKÜ
BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ

Kaynaklar

- **Java'da Ternary Operatörü Anlatımı İçin;**
- http://mehmetduran.com/Blog/Makale.html/Ternary-Operator-Kullanimi-(kosul--dogru---yanlis)/336
- https://aydincanaltun.com/ternary-uclu-operator/
- https://www.programiz.com/csharp-programming/ternary-operator
- https://www.yazilimhocasi.net/Makale-97-c
- https://www.javatpoint.com/ternary-operator-in-java
- https://www.geeksforgeeks.org/java-ternary-operator-with-examples/
- https://www.tutorialsteacher.com/csharp/csharp-ternary-operator
- https://www.tutorialspoint.com/Java-Ternary-Operator-Examples

YouTube BM DERSLERİ

http://youtube.com/bmdersleri

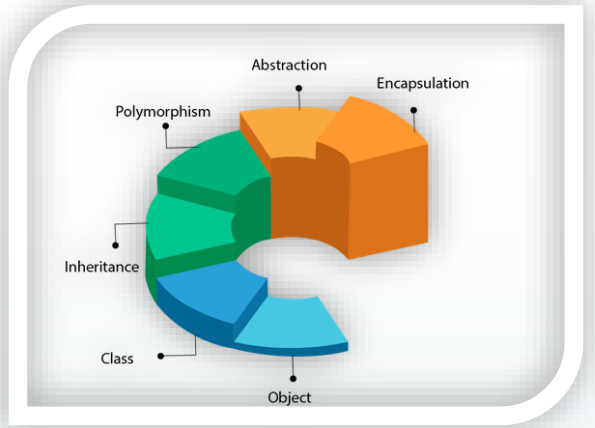
YouTube BM DERSLERİ

http://youtube.com/bmdersleri

MAKÜ

BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ

Nesneye Dayalı Programlama Dersi



İlginiz için teşekkürler…

Hazırlayan ve Sunan : **HİLAL ÇINAR – 1911404044**

E-posta : hilallcinarrr@gmail.com

Tarih : 11/06/2021

Sürüm : v2

Ders Yürütücüsü : Doç. Dr. İsmail KIRBAŞ